	Type	L #	Hits	Search Text	DBs	Time Stamp
1	IS&R	L1	883	((345/440) or (345/440.1) or (702/67) or (702/68)).CCLS.	USPAT	2002/04/30 12:30
2	BRS	L2	258	1 and event	USPAT	2002/04/30 12:09
3	BRS	L3	1	1 and event adj table	USPAT	2002/04/30 12:10
4	BRS	L4	18	1 and glitch	USPAT	2002/04/30 12:09
5	BRS	L5	4	1 and event same trend	USPAT	2002/04/30 12:09
6	BRS	L6	398	event adj table	USPAT	2002/04/30 12:11
7	BRS	L7	1	event adj table same trend	USPAT	2002/04/30 12:14
8	BRS	L8	2	event adj table same trend	EPO; JPO; DERWENT; IBM_TDB	2002/04/30 12:14
9	IS&R	L12	1	("ep-508386-A2").PN.	EPO	2002/04/30 12:16
10	IS&R	L13	1	("ep-434050-A2").PN.	EPO	2002/04/30 12:23
11	BRS	L14	60	1 and log	USPAT	2002/04/30 12:24
12	BRS	L15	24	1 and log and (gui or user adj interface)	USPAT	2002/04/30 12:29
13	IS&R	L16	226	(345/440.1).CCLS.	USPAT	2002/04/30 12:30
14	IS&R	L17	150	((702/67) or (702/68)).CCLS.	USPAT	2002/04/30 12:30

09/378,969

DOCUMENT-IDENTIFIER: US 5768148 A
TITLE: Man machine interface for power management control systems

---------- KWIC ---------

DWKU:
5768148

ABPL:
A utility for rapid development of three dimensional representations of
electrical distribution switchgear is provided. These switchgear elevations
have logical connections to the switchgear devices. An elevation can be
modified to any dimensions with an infinite number of combinations and
arrangements of meters and protection devices to quickly and accurately
represent a customer's switchgear. Also, an event logger utility is provided
for viewing, organizing and analyzing unusual behavior in a power system. The
event logger utility passes a received message as an un-acknowledgeable or
acknowledgeable alarm or as an event based upon the contents of a
initialization file. A utility for the waveform capture is provided for
viewing and analysis of waveforms (e.g., Fourier, frequency and/or harmonic
analysis) captured by sophisticated metering devices. Waveforms from a device
may be super-imposed for analysis. Processing of the collected waveform data
to display any one of the eight waveform parameters (i.e., I.sub.a, I.sub.b,
I.sub.c, I.sub.n, V.sub.a, V.sub.b, V.sub.c, V.sub.x) or a combination thereof
including an "all" selection in a window tiling format is provided.

BSPR:
Software for monitoring and controlling selected aspects of

power
usage/consumption is loaded into the computer as described above and includes a
dynamic data exchange (DDE) server. The DDE server allows external programs to
access power management data in a Microsoft Windows environment. Data
interface to the DDE server is provided by the system through the Wonderware
Intouch utility or any other DDE aware program. The DDE server configuration
and control interface is provided through DDE server application window menus.
The DDE server supports DDE aware clients such as Excel or other modules, which
include a waveform capture module, an event logger module, productivity
modules, and a Wonderware Intouch module. The Wonderware Intouch module
includes a tool kit for building screens and interfaces, and a graphical user
interface for monitoring and controlling the electrical distribution system.

BSPR:
The event logger module provides for viewing, organizing and analyzing unusual
behavior in a power system. The event logger module includes a utility that
passes a received message as an un-acknowledgeable or acknowledgeable alarm or
as an event based upon the contents of an initialization file. Electrical
meters and control/protection devices use various codes to describe occurrences
to the circuits that are monitored or controlled. A file collates these codes
into three categories for analysis. These three categories for any particular
device are modifiable for the code received from a device.

BSPR:
The DDE server provides a mnemonic cross reference between register items and
standardized, alphanumeric parameter names. This mnemonic interface allows the
user to retrieve data from a device without knowledge of the actual device

register item number. The DDE server further provides a consistent device event data item for dissimilar devices. Also, the DDE server automatically performs time synchronizing for all supported devices and provides a consistent waveform interface. The DDE server is optimized for either the Modbus RTU or Ethernet protocol.

DEPR:
Referring now to FIG. 4, a block diagram of the software for monitoring and controlling selected aspects of power usage/consumption, discussed above is generally shown. This software is loaded into the computer as described above and includes a dynamic data exchange (DDE) server 152. DDE server 152 allows external programs to access power management data in a Microsoft Windows environment. Data interface to DDE server 152 is provided by the system through a Wonderware Intouch utility. The DDE server is a 16 bit application under Windows NT. A configuration and control interface for the DDE server is provided through server application window menus. Associated with DDE 152 are logical data tables 154 and related modules, i.e., an Excel or other DDE aware applications module 156, a waveform capture module 158, an event logger module 160, productivity modules 162, and a Wonderware Intouch module 164. Module 164 includes a tool kit for building screens and interfaces, and a graphical user interface 164a for monitoring and control of the electrical distribution system. The graphical user interface 164a for the server operates in 32 bit Windows or Windows NT environment and Intouch library functions. Module 158 provides for viewing and analysis of waveforms (e.g., Fourier, frequency and/or harmonic analysis) captured by sophisticated metering devices. Module 160 provides for viewing, organizing and analyzing unusual

behavior in a power system. Productivity modules 162 include, for example, a cost allocation module and a load management module. The cost allocation module provides for tracking power consumption to the sub-unit level, developing internal billing methods and reports, thereby reducing cost. The load management module provides for tracking power demand and automatically shedding non-critical loads to prevent peak demand penalties, and provides for timer-based control to reduce power consumption. DDE server 152 communicates though the interface card, i.e., the RS485 interface cards 124, or a RS232 to RS485 convertor, in the embodiment of FIG. 2 and the Ethernet interface cards 144 in the embodiment of FIG. 3.

DEPR:
The event logger module 160 includes a utility that passes a received message as an un-acknowledgeable or acknowledgeable alarm or as an event based upon the contents of an initialization file. The DDE server ensures that all events are cast in the same format so that the event logger module 160 can interpret each event. Electrical meters and control/protection devices use various codes to describe occurrences to the circuits that are monitored or controlled. A file collates these codes into three categories for analysis. These three categories for any particular device are modifiable for the code received from a device. Referring to FIG. 5, the utility accesses codes from various types of devices and determines, via a look-up in the file, which of three categories this code shall be reported by. The three categories are `ACK/UNACK` for acknowledgeable alarms, `-` for alarms requiring no acknowledgment, and events which are merely reports of device status. The key feature is the

initialization file which allows new devices to be added or the category for an existing device's code to be modified without rewriting the utility.

DEPR:
DDE server 152 is described hereinbelow for the RS485 interface embodiment. However, it will be appreciated that the same is applicable to the Ethernet interface embodiment with the exception that the server is optimized for Ethernet protocol (instead of Modbus RTU protocol). DDE server 152 provides a mnemonic cross reference between Modbus RTU register items and standardized, alphanumeric parameter names. This mnemonic interface allows the user to retrieve data from a device without knowledge of the actual device register item number. DDE server 152 further provides a consistent device event data item for dissimilar devices. Also, DDE server 152 automatically performs time synchronizing for all supported devices and provides a consistent waveform interface. DDE server 152, for the RS485 interface, is optimized for Modbus RTU protocol by compensating for invalid ranges of registers in the device polling packets and it also provides superior protocol debugging capabilities by displaying the complete Modbus RTU input/output packet traffic. In addition, the DDE server 152 performs automatic conversion between 16 bit and 32 bit device register modes. A generic interface allows any Modbus RTU device to be used with the system. The DDE server uses the Modbus RTU protocol standard for communication to metering, relaying and I/O devices using the RS-485 communication ports.

DEPR:
By way of example, several of the devices discussed above, i.e., EPM, RMS6, RMS9B, RMS9C, RMS9D, ECM, MDP and the Modbus concentrator,

include register types R3XXXX; R4XXXX; and R0XXXX. To acquire data from these devices and to download set points, the DDE server implements Modbus function codes, 01, 03, 04, 05, 06, 16, 56. Various register groups, i.e., dynamic values, set points, command registers, event registers, fixed value registers, and Commnet statistics can be configured by the user into either Slow Poll Register or Fast Poll Registers or Poll Once Registers. The DDE server downloads time stamps to all these devices periodically. Time download periodicity is available in an initialization file. An event count register is polled at the Fast Poll rate. When the event count register is non-zero, an event fetch is performed by reading the start address from the event buffer. Each event is read and time stamped sequentially. The event code is expanded with a description, additional data fields, and a date stamp of the events before sending to client.

DEPR:
The Multilin 269 device, discussed above, includes register types R3XXXX and R4XXXX. To acquire data from this device and to download set points, the server implements Modbus function codes, e.g., 03, 04, 16. This device has no time or event registers.

DEPR:
The Multilin 565 device, discussed above, includes register types R3XXXX; R4XXXX; and R0XXXX. To acquire data from these devices and to download set point, the server implements Modbus function codes, e.g., 03, 04, 05, 06, 16. Date and time stamped event strings formed from buffered events are provided.

DEPR:
The PML 3710 device, discussed above, includes register

type R4XXXX. To
acquire data from this devices and to download set points, the server
implements Modbus function codes, e.g., 03, 16. The waveform capture feature
of PML 3710 is supported by the DDE server. Date and time stamped event
strings formed from buffered events are provided.

DEPR:
The PML 3720 device, discussed above, includes register type R4XXXX. To
acquire data from this device and to download set points, the server implements
Modbus function codes, e.g., 03, 16. The waveform capture and waveform
recording features of PML 3720 are supported by the DDE server. Date and time
stamped event strings formed from buffered events are provided.

DEPR:
Referring to FIGS. 33 and 34, the device module 202 abstracts a field device
and handles multiple register categories (or register groups). Class,
CCoilItem, provides support for the special discrete register bit operations
for R0 and R1 reference register types. Class, CDev, contains all device
information like device name, device ID, device type name and com-port address.
Class, CDeviceType, abstracts a field device type the server will talk to, has
a variable number of register groups, contains a mnemonics list, and has
function codes associated with it. Class, CDiscreteItem, extracts discrete bit
array in a register, for a 16-bit register, all 16 bits can be programmed while
for a 32-bit register only the lower 16 bits can be specified by user. Class,
CEventItem, handles the event processing for a selected device, e.g., a General
Electric Co. device, formats an event string and pass it to the client.
Class, CGEDeviceType, supports event register group, by way of example, part

nos. EPM, RMS6, RMS9B, RMS9C, RMS9D, ECM, PLMeter and MDP qualify as General
Electric Co. device types. Class, CIntItem, takes care of unsigned and signed
integer items. Class, CItem, is a data point present in field device. CItem
comprises a plurality of registers in the device and can handle a single type
or array of types. CItem is the base class for different types of items
present in the application and include integer, long, real, coil, discrete bits
and real time parameters. An item has reference to the register group it
belongs to. Whenever a new item is created or activated a dummy register block
is created and polled immediately for the fast updating of the item. Class,
CLongItem, handles a signed long data type. Class, CMnemonic, contains the
mnemonic name and the corresponding register format. Class, CModbusConcType,
automatically synchronizes time with all Modbus concentrator device which have
only time download property. Class, CMultilin565EventItem, handles the special
event processing present for Multilin 565 device. Class, CMultilin565Type,
supports event register group and processing for Multilin meters, which are
similar to that of General Electric Co. devices. Class, COnLineDevice, exist
only when the server is running, whenever an on-line device is created, a copy
of its device type is created and attached to the on-line device. Class,
CPLMeterType, is associated with the General Electric Co. Power Leader meter
devices and supports CPLMeterWFCReg register group for waveform capture.
Class, CPLMeterWFDataItem, is a text item associated with the PLMeter.
CPLMeterWFData item is a collection of `n` samples read from the device, with
the number of samples and sample's start address are read from the
application's .INI file. Class, CPML3710EventItem, handles the event

processing for a PML3710 Device and formats an event string and passes it to
the client. Class, CPML3710Type, supports a PML3710 Device where the default
register group CPML3710WFCRegGroup supports waveform capture. The event
processing is similar to other device types except that it doesn't contain an
event count register. Class, CPML3710WFCDataItem, is a text item associated
with the PML3710. CPML3710WFCDataItem is a collection of `n` samples read from
the device. Each sample is 12 bits and two consecutive registers, giving three
values. The number of samples and sample's start address are read from the
application's .INI file. Normally for PML devices the WFC is 99 registers.
Class, CPML3720AvailItem, handles the PML.sub.-- WFC.sub.-- AVAIL special item
in the PML3720 device. Class, CPML3720EventItem, handles the event processing
for a PML3720 Device and formats an event string and pass it to the client.
Class, CPML3720Type, supports a PML3720 device. The default register groups
that go with this device type are CPML3720WFCRegGroup and CPML3720WFRRegGroup
for handling waveform capture and waveform recording respectively. Class,
CPML3720WFCDataItem, is a text item associated with the PML3720.
CPML3720WFCDataItem is a collection of `n` samples read from the device, where
each sample is 12 bits and two consecutive registers, giving three values. The
number of samples and sample's start address are read from the application's
.INI file. Class, CPML3720WFRDataItem, collects required sample data and
updates them to the client. CPML3720WFRDataItem handles the PML.sub.--
WFR.sub.-- BFRS.sub.-- AVAIL and PML.sub.-- WFR.sub.-- NEXT.sub.-- BFR special
items in the PML3720 device. Class, CRealItem, keeps data in an IEEE floating
point format, in 16-bit mode, two registers are required while in 32-bit mode,

only one register will give the value. Class, CStatusItem, is a text item
which exists in every device. The status strings that are updated by this item
to a device are ACTIVE and DEAD.

DEPR:
The Normal State Flag is a flag indicating that the main processor is in the
Normal state. The following tasks is performed when the main processor is in
the Normal state: self-test and initialization; address conflict resolution;
user interface; LED display; key-pad interface; BIT diagnostics; SCI interface;
transmit; receive; SCI (RS-485 Modbus) interrupt service; IPC interface; poll;
command; IPC interface interrupt; service; event handling; and time
synchronization.

DEPR:
In polling, the segment processors poll the Commnet devices connected to the
Modbus concentrator, process and forward the received data from the Commnet
devices to the main processors IPC task handler so that the received data can
be placed in a memory buffer transfer to the Computer. Commnet devices are
polled according to a Commnet address table. Commnet device polling is having
the segment processors send an update request to Commnet devices for the
Commnet devices to respond with the contents of their registers. The data sent
to the segment processors include the dynamic value registers, fixed value
registers, set point registers, event registers and the statistics registers
contents. On receipt of this data from the segment processors, the main
processor's IPC task unpacks the data and place it in the IPC receive data
buffer. The segment processors service the main processor's request, obtain
the Commnet device register data and forward the Commnet register data to the

main processor.

DEPR:
An event is defined as an asynchronous condition occurring outside the normal operation of a Commnet device. All asynchronous events is logged by the Modbus concentrator. The main processor maintains a global event table for all Commnet devices. The main processor polls the segment processors for Commnet device status and on receipt of the response to this request, the main processor places all event logs received into a global event table for the system. The most recent 24 events is maintained and be available to the computer from the main processor. The main processor stores event logs in the chronological order in which they were received with EVENT 1 being the oldest event and EVENT N being the latest one. The main processor clears the global event file of all events transferred to the computer. Each event message is time stamped with a 10 mil. sec. accuracy. Events is stored in the same format for all Commnet devices. The main processor maintains the number of active events in the Active event register.

DEPR:
The IPC Function 03 Response, for each of the segment processors develops and returns an IPC Function Response to IPC Function 03 to the main processor in response to receiving an IPC Function Request 03, Read R/W Registers, from the main processor. The RS-485 Function Request 3, RS-485 request from the computer to the main processor. Function Request defines number of event logs to be transferred to the Computer from the main processor. The main processor maintains the global event file and transfer event logs to the computer on request from the computer.

DEPR:
The event file stores between 0 and 24 event logs. If there are more events
than can be stored in the event file, the oldest event is removed and the rest
of the events in the event file are shifted downward, freeing the event log for
the earliest event entered. The event logs is automatically cleared after the
computer reads them. If the computer does not read all the recorded events,
the event registers are shifted to represent remaining events to be read by the
computer. The number of active events reflects unread events by the computer.
The computer may read event registers at any event log boundary. Event
registers following the event that was read is shifted up to fill in the event
logs transferred to the computer. Event logs before the event logs that are
read will not be affected. Computer request to read event registers partially
will be returned with an exception code.

DEPR:
The main processor polls the segment processors for their Commnet device status
through IPC Function Request 03. When the main processor receives the status,
the main processor places the event log received into the global event file in
the device register buffer and increment the number of events contained in the
Active event register. The main processor always add new event logs to the
head of the global event file in the device register buffer.

DEPR:
The global event file is cleared on the application of power to the Modbus
concentrator and is maintained while power is applied to the Modbus
concentrator.

DEPR:
The IPC Function Request 03, IPC Read R/W Registers request

from the main
processor to each of the segment processors. Function Request 03 is the
request for the segment processors to transfer the contents of their data
registers to the main processor. The Function Request 03 Response, the main
processor returns to the computer the RS-485 bus address, function code 03,
starting memory address and quantity of data registers to be transferred to the
computer and the contents of the data registers in response to receiving RS-485
function request 03 from the computer. The Active event register is the number
of active events in the global event file, ranges between 0 and 24. The Dev
reg buff is a Commnet device register file.

DEPR:
Each of the segment processors inter-processor communication path with the main
processor is checked by having the segment processors receive a test-telegram
from the main processor and then sending the appropriate response frame back to
the main processor. Each of the segment processor interfaces and their
peripherals (SPI, etc.) and all internal and external RAM for start of
processing are initialized. The segment processors timer is initialized. The
timer is initialized to 4096 counts (2.22 mil. seconds time intervals). The
processors address table is initialized. The main processor sends the set
device address request to each of the segment processors for initialization of
all Commnet devices connected to the Modbus concentrator. On receipt of the
Preset Multiple Registers request frame, the segment processors clears (set to
0) the set point registers that contain the event time (hr., min., sec.)
reference. The event time reference starts incrementing after initialization
is complete. The segment processors enters the Idle state when they receive an

IPC message frame that commands the segment processors to enter the Idle state when the main processor declares that it has a non-recoverable error or by the segment processors not receiving any message frames from the main processor. In case of an unsuccessful self-test, the segment processor disables all its functions except for responding to a diagnostics byte request from the main processor and turn on its red LED to indicate that it has a non-recoverable error. On successful completion of the self-test, the segment processor initializes its peripherals (SPI, etc.) and internal and external RAM for normal operation and turn its green LED to indicate that the segment processor has no initialization self-test errors.

DEPR:
The Normal State Flag indicates the segment processor is in the Normal state. The following tasks are performed when the segment processor is in the Normal state: self-test and initialization; BIT diagnostics; segment auto configuration/address conflict resolution; LED display control; IPC interface; transmit; receive; IPC interface service; HBPC kernel--Commnet interface; data collection; engineering unit conversion; execute control commands; event handler; and time sync.

DEPR:
Data collection is the process of the segment processors polling the Commnet devices connected to the segment processor with register read commands to obtain the contents of the Commnet devices dynamic value registers, fixed value registers, set point registers, event registers and statistics registers. On receipt of the response frames from the Commnet devices, the segment processors places the Commnet devices register contents into the appropriate device

register data list in the segment processor's device register buffer. The
Commnet devices registers is polled in the sequence the device registers are
contained in the devices register address table. The received data is retained
in the binary format (raw data) it is received in.

DEPR:
The segment processor develops Commnet device request frames and send the
request frame to all of the Commnet devices connected to the segment processor.
On receipt of the Commnet device request frame, the Commnet device places the
contents of the particular dynamic value register, fixed value register, set
point register, event register or statistics register addressed into a response
frame and return the response frame to the segment processor. The HBPC kernel
removes the message frame overhead from the response frame and place the
received register data into the HBPC kernel's receive data buffer. The Data
Collection task then places the register data into the associated devices
register data list in the device register list. The register data is retained
in the binary format (raw data) it is received in. The binary form of the
register data is: Bnn/Bmm, where 0 nn 24,0 mm 24 and nn mm.

DEPR:
The Commnet dvce frame is a Commnet device request frame that identifies a
Commnet device's dynamic value register, fixed value register, point register,
event register and statistics register contents are to be transferred to the
segment processor in the devices Commnet response frame. The Device raw reg
lst is a list of a devices raw register data received from the Commnet devices,
contained in the segment processors device data buffer.

DEPR:
The segment processors polls the Commnet devices connected

to the segment
processor for the data contained in the dynamic value registers, fixed value
registers, set point registers, event registers and the statistics registers
and on receipt of this data places the event register data into event logs.
The segment processor maintains the event logs in a linked list in the IPC
transmit buffer. On receiving Function Request 03 from the main processor, the
segment processors transmits the event logs to the main processor in the
response to IPC Function Request 03. The IPC Function Request 03, IPC Read R/W
Registers request from the main processor to each of the segment processors.
The Function Request 03 is the request for the segment processors to transfer
the contents of their data registers to the main processor. The Commnet
Response frame 84, a response frame received from Commnet devices to obtain the
data contained in the dynamic value registers, fixed value registers, set point
registers, event registers and the statistics registers.
The Dev addr table is
a table of addresses of Commnet devices connected to the Modbus concentrator.
The Cmnt Dvce Dat is received data/value frames from Commnet devices (received
in HBPC kernel receive buffer) and which are transferred to the segment
processors ipc tx buff for transfer to the main processor.
The Wall time is
the current time.

DEPR:
The segment processors polls the Commnet devices through the Commnet function
request 84 (Update Registers) for the Commnet devices the dynamic value
registers, fixed value registers, set point registers, event registers and the
statistics registers data employing the Commnet device address table. After
the response frames are received, the segment processors places the devices

event register data into the device event logs in the IPC transmit buffer. The segment processor adds the current time to the devices event logs. The event logs is maintained in a linked list in the IPC transmit buffer. The main processor polls the segment processors for their Commnet device status through IPC Function Request 03. When the segment processors receive this request, the segment processors develops the IPC Response frame for IPC Function 03 and transfer the its event log to the main processor. The segment processor clears all event logs transferred to the main processor after the event logs are transferred to the main processor. The segment processor maintains event logs in the IPC transmit buffer. The Commnet Request frame 84, an update register request frame sent to all Commnet devices to obtain the data contained in the dynamic value registers, fixed value registers, set point registers, event registers and the statistics registers. The IPC Function 03 Response, each of the segment processors develops and returns IPC Function Response to IPC Function 03 to the main processor in response to receiving a IPC Function Request 03, Read R/W Registers, from the main processor. The ipc tx buff, Commnet device received register data/value, placed into IPC transmit buffer for transferring device status to the main processor.

DEPR:
The DDE server simulator simulates the modbus register maps of the electrical distribution, monitoring, and control devices. The DDE server simulator also uses the same base configuration as the DDE server (described hereinbefore) and simulates the behavior of a number of communicating devices without being connected to a network. Some of the key features of DDE server simulator are:
user configurable power simulation profiles and topic setup

parameters;
mnemonic cross reference between register items and
standardized, alphanumeric
parameter names; consistent device event and trip
simulation for dissimilar
devices; consistent waveform analysis interface and
simulation between
dissimilar devices; and generic support for any Modbus RTU
compliant device.

DEPR:
The user is required to input event code to generate an
event. The DDE server
simulator supports event generation for the device types
with event definition
in register map.

DEPR:
Programmed parameters are either values configured by user
or setpoints
downloaded from a DDE client. When simulation for measured
parameters is
"frozen", all parameters are programmed parameters until
simulation is resumed.
All registers of the simulated register map of a device are
available for
viewing by the user. Where a register allows modification
(read/write or write
only), the user can also modify contents of that register
through the device
simulation interface. The DDE server simulator allows the
user to generate
abnormal field conditions such as relay trip condition,
event occurrence,
clearing measured/computed parameters etc., through an
interface provided by
device simulator screen. A list of measured, computed and
programmed
parameters for each device is provided below.

DEPR:
Using the event code as input by the user, the time stamp
is applied, the event
code loaded into the proper register and the event counter
register is updated.
When the event link is established, the simulator supplies
the appropriate
event item. The number of event registers supported for

devices is the same as
the ones supported by individual devices (i.e., as specified in the
corresponding register maps). On user request for trip simulation,
corresponding register values are set to simulate trip conditions.

DEPR:
The DDE simulator provides the updated measured, computed and programmed
parameter data to the DDE clients upon request. The DDE server simulator also
retrieves commands from DDE clients and forms the proper string from register
data and supply data under the appropriate waveform data item. Additional data
to be updated on request includes appropriate setpoint registers in response to
setpoint download requests from client, waveform record/capture data to DDE
clients and all register values in the device register maps to DDE clients.
The user will be able to access simulated data points of DDE link using
register address of items. Only mnemonics can be used in the case of special
items such as event, status etc. Additional outputs include: register contents
on device simulator screen; instantaneous values of parameters of selected
device on device simulator screen; profile being used for simulation of
measured parameters along with superimposed noise on device simulator screen;
simulator configuration handling; data value update to clients for active
topics and items; configuration dialog box handling for user requests to
configure server simulator; server simulator execution and termination on user
request from main menu.

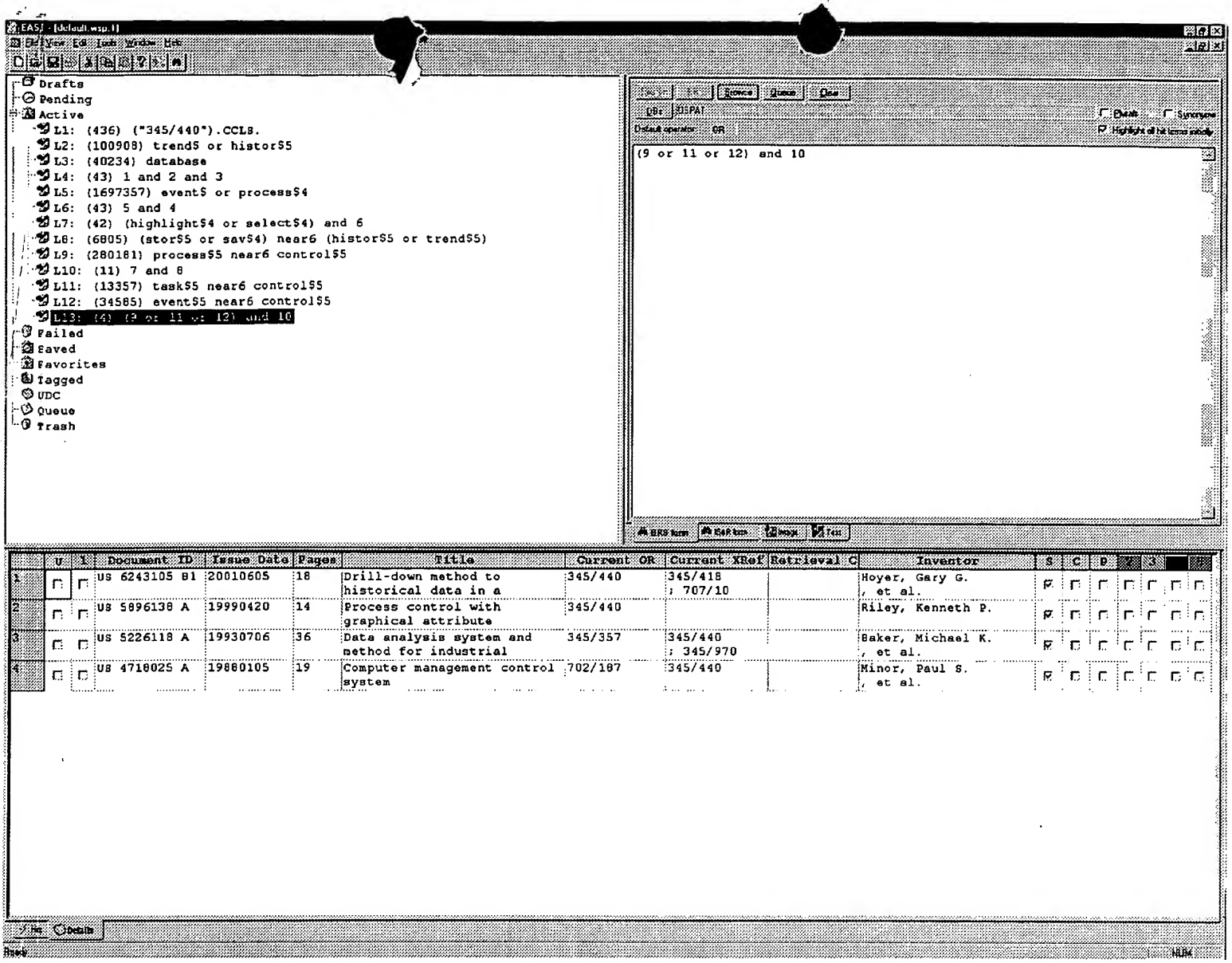
DEPR:
The DDE server simulator will support event generation for the device types
with event definition in register map. A user interface is provided for event

generation through which the user may input/select an event code. Time
stamping of event, loading the event code into the proper register and
maintenance of event counter register is performed by the DDE server simulator.
When the event link is established, the DDE server simulator shall supply the
appropriate event item. The number of event registers supported for devices
shall be same as the ones supported by individual devices (i.e., as specified
in the corresponding register maps).

DEPR:
The user interface includes dialog boxes allow selection of profile parameters
and noise characteristics of data values for each measured parameter. Also,
any parameter that needs accumulation of values (e.g., energy) is generated
using an increment value defined in the profile dialog boxes. The
configuration dialogs are the same as that for DDE server. The topic
configuration file formats are the same as that of respective DDE server
configuration data file. Event generation and trip simulation are established
as discussed above. A user interface is also provided for command coils of
General Electric Company devices.

DEPR:
The data handler module 72 is an on-line module which handles all register map
data for all the data being simulated and performs the following functions:
establishes the register map for all devices being simulated; permits reading
and writing of the register maps; performs event handling for all devices which
supports events; performs command coils specific functions; triggers waveform
commands to the user I/F handler module 73; saves register map data to files;
and handles data in 32-bit format for PML 3710 and 3720.

DEPR:
The user I/F handler module 73 is an on-line module which handles user I/F for
device simulation. The user I/F handler module 73 performs the following
functions: setup all devices configured at server for simulation; provides user
interface for event generation; provides user interface for special commands
such as relay tripping, clear energy etc, generates data for measured and
computed parameters using profiled data (sinusoidal, saw tooth, triangular,
pulse, step profile) for user defined time intervals; provide user interface
for viewing or changing register values for all type of devices; handles 32-bit
mode processing for PML 3710 and 3720; user selected parameter is plotted
graphically; displays measured and computed parameters instantly; allows
freezing and unfreezing of simulation; and performs waveform capturing and
recording.



09/378,969
Havekost et al.